A Scene-based Augmented Reality Framework for Exhibits

Julien Li-Chee-Ming, Zheng Wu, Randy Tan, Ryan Tan, Naimul Mefraz Khan, Andy Ye, Ling Guan
Ryerson Multimedia Research Laboratory, Ryerson University, Toronto ON M5B 2K3, Canada

Abstract. This paper presents a novel augmented reality (AR) framework specifically targeting scene-based exhibits. Unlike traditional AR libraries that rely on specific image targets, the proposed framework utilizes a bag-of-words model for scene recognition, which enables recognition and subsequent launch of AR experiences under challenging environments. Moreover, the proposed framework utilizes relocalization capabilities of the popular ORB-SLAM algorithm to track the user's movement after recognition of a particular exhibit. We demonstrate the efficacy of the proposed framework through a complete mobile app designed for a local museum, where artifacts are enhanced through AR.
Keywords: bag-of-words, SLAM, object recognition, scene recognition.

1 Introduction[1]

Augmented Reality (AR) is a live view of a real-world environment whose elements are augmented by computer-generated content such as text, sound, video, or animation. Recently AR has become a ubiquitous mode of application development due to the adoption of new technologies, such as mobiles devices, computer vision, 3D rendering, image recognition, and cloud computing. The commercial market for AR applications is projected to increase from $5.2 billion in 2016 to more than $162 billion in 2020 [1]. In an exhibit scenario, such as a museum or a gallery, AR has huge potential to increase visitor engagement. Instead of reading static text or visuals that typically accompany such exhibits, an AR solution can present dynamic content in a context aware manner with accurate registration and tracking.

This work presents a novel framework for accurate scene recognition and tracking specifically geared towards creating AR exhibits for museums. The primary contributions of this work can be summarized as follows:

1. *Perform fast and accurate scene recognition on mobile devices*: The initialization of a good AR experience for an object on display greatly depends on accurately recognizing the 3D object so that virtual content can be retrieved and aligned with the real-world view through the device's camera. Unlike traditional image target-based approaches for initialization that is prevalent in commercial SDKs such as Vuforia [2], ARCore [3], ARKit [4], we utilize a 2D to 3D matching algorithm that is tradi-

[1] NSERC's financial support for this research work through a Collaborative Research and Development (CRD) Grant (#507333-16) is much appreciated.

tionally used for scene recognition. We show that for challenging AR initialization scenarios, the scene recognition algorithm succeeds while image-based targets fail.
2. *Perform robust real-time tracking on mobile devices*: For a smooth AR experience, real-time tracking is essential once the AR content is retrieved and initially aligned. A state-of-the-art Simultaneous Localization and Mapping (SLAM) algorithm [5] is utilized to work with the recognition module in delivering a seamless AR experience. To further speed up tracking, the mobile devices' Inertial Measurement Units (IMUs) are utilized to interpolate orientation between SLAM-based tracking calls.

Using the algorithms, a prototype application was developed for the newly renovated Canada Science and Technology Museum (CSTM), where various museum artifacts were recognized, and highly immersive educational experiences were initiated based on recognition and tracked in real-time using SLAM.

2 Related Work

The usage of AR in exhibits has recently attracted some attention. [6] promoted MRsionCase, a device-free mixed reality (MR) showcase that presents spatially consistent visual and auditory information with physical exhibits. However, the proposed system displays only 2D images at a fixed position, which lacks depth perception and interactivity. [7] built a museum display in the Mawangdui Han Dynasty Tombs and overlaid virtual archaeological relics onto real exhibits. The Vuforia SDK [2] was used to perform object recognition and camera motion tracking. Although the AR experience did not require external infrastructure, it needed a calibration grid in the mapping stage that must be approximately the same size as the object being mapped. This process becomes cumbersome with large artifacts. [8] developed a museum tour guide application which gives users access guiding services and other functions interactively. The proposed navigator relies on pre-installed sensor beacons for localization. [9] is our own previous work, where we developed a large-area augmented reality exhibit for the Fort York National Historic Site, where multiple users can experience the same shared events simultaneously. We utilized specialized hardware that performed radio-frequency-based tracking. The work presented here demonstrates that a similar experience can be achieved using only a mobile device's camera.

3 Background

A key challenge in an AR-based exhibit design is localizing the user with respect to the world. Typically, image-based targets are used in commercial libraries such as Vuforia [2], ARCore [3], ARKit [4], where a 2D image is utilized at the recognition stage to localize the user. However, in challenging scenarios such as low light, distance from 2D image, such recognition fails, as we see in the experimental results section. Another alternative is to entirely depend on relocalization with SLAM. However, as we show, the relocalization approach is prone to drift due to accumulated errors and it is difficult to achieve real-time performance with acceptable accuracy on mobile devices. Although ARCore/ARKit has good native implementation of SLAM,

none of the libraries have consistent relocalization, and the libraries only support newer devices. Instead, we approach the problem in two stages: 1) 3D scene recognition; where we utilize a bag-of-word models to recognize the current exhibit instead of the traditional 2D image approach for robust recognition; and 2) camera motion tracking; where we enable relocalization within SLAM to track the user's movement.

Decoupling the recognition from tracking ensures consistent launch of AR experiences followed by tracking. Any potential drifts/slowdown arising from SLAM does not affect the initialization of the experience. The bag-of-words model provides instantaneous and robust recognition, which enhances user experience.

3.1 Scene Recognition

[10] surveyed methods for scene recognition and argued that appearance-based image-to-image matching techniques perform better than map-to-map and image-to-map methods. Within appearance-based methods, the bag of words methods (BoW) have the highest performance. [11] proposed the bag of binary words obtained from BRIEF descriptors along with the very efficient FAST feature detector, reducing the time needed for feature extraction by more than one order of magnitude. However, BRIEF is not invariant to rotation nor scale, which limited the system to in-plane trajectories and loop closures with similar point of view. [12] extend that work using ORB, which are rotation invariant and can deal with changes in scale. The bag of binary words method DBoW2 [11] is used for image to image matching mainly for the purpose of scene recognition inside SLAM algorithms [12]. Scene recognition within SLAM algorithms are used for relocalization and loop closing. Relocalization uses image to image matching to allow for SLAM algorithms to recover from tracking failure which would render previously mapped data useless in visual odometry algorithms. Loop closing reduces the odometry drift error by detecting paths to revisited areas and then minimizes the error over the path. This work will apply DBoW2 to recognize the artifact being viewed by the camera and trigger its AR experience.

3.2 Camera Motion Tracking

From an AR context, tracking based on visual features mostly use some variant of Simultaneous Localization and Mapping (SLAM). The objective of SLAM in an AR context is to anchor the virtual objects to their position in the real world, regardless of the motion of the user. To accomplish this, the system must be able to track the camera pose (i.e., 3D position and orientation) relative to the anchor point. The chosen SLAM system for this work is ORB-SLAM [5], which runs three parallel threads: tracking, mapping, and loop closing. ORB-SLAM uses ORB features in their localization, tracking, mapping, and loop closing. The ORB feature allows the algorithm to have real-time performance without relying on GPUs and provided some robustness to lighting changes and invariance to rotation and scaling. All the functions using the same descriptor also allows ORB-SLAM to be efficient in both memory and execution time by preventing the need to compare different descriptors. ORB-SLAM uses DBoW2 when performing relocalization and loop closure detection. These functions

allow ORB-SLAM to have more re-usability and more robustness while maintaining its real-time performance.
The tracking thread maintains camera localization and determines when a new keyframe should be added. The local mapping thread oversees the mapping of the keyframes. The loop closing thread then takes the new keyframe and map points to calculate a similarity score to its closest matches using DBoW2. If the frames pass a threshold then the thread will attempt to minimize the mean square error of the entire loop rather than minimize local error.

4 Methodology

This section explains how the scene recognition module and the tracking module are integrated to deliver a novel AR experience for museums. Four modules were developed (Fig. 1): An offline module to perform training for scene recognition; an offline module to generate and save a map for each artifact; an online module to perform scene recognition; and an online module to perform motion tracking.
ORB-SLAM was selected to perform camera motion tracking in Modules 2 and 4, as it has been shown to perform well on mobile devices [13]. DBoW2 was selected to perform object recognition in Modules 1 and 3 because it has also been shown to performance well on mobile devices [14]. Further, DBoW2 is used for ORB-SLAM's relocalization and loop closing, which facilitated the integration of the modules. The software libraries were built for Android and integrated as Unity 3D game engine plugins. Rendering the AR content was also performed by Unity 3D.

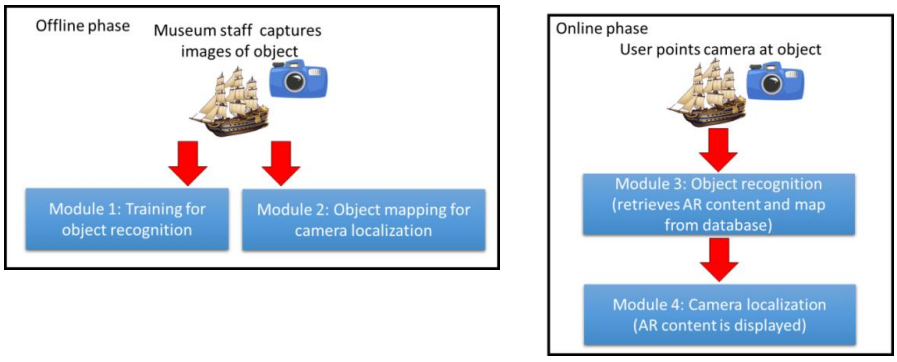

Fig. 1. The two phases performed by the museum staff and the user respectively

4.1 Module 1: Training for scene recognition

Training for scene recognition is an offline process performed by the museum staff, which involves capturing images of each artifact from various vantage points. These vantage points are predictions of how the user will view the artifact with their mobile device. Images were captured using various image resolutions and aspect ratios, and

from various heights, viewing angles, and distances from the artifact. These images were processed using DBoW2. 10 to 20 images were captured per artifact, as this number produced a sufficiently small database size, while maintaining high precision and recall rates for object recognition. A verification step was developed to further increase the precision and recall of DBoW2. Specifically, DBoW2 returns the object and image with the highest match score. An object was accepted if it was returned 7 times among 10 consecutive images, as this ratio provided an acceptable balance between reliability and real-time performance.

The visual vocabulary was created by [12] in an offline step with the dataset Bovisa 2008-09-01 [15]. This dataset is a sequence with outdoors and indoors areas, yielding a vocabulary that provides good results in both scenarios. ORB features were extracted from ten thousand images from the dataset and a vocabulary of 6 levels and 10 clusters per level was built, getting one million words. Such a big vocabulary is suggested in [11] to be efficient for recognition in large image databases.

The vocabulary had a file size of 43 MB. The training database generated by DBoW2 had a file size of 14 MB. Both files are stored onboard the mobile device when the application is installed. On a Google Pixel, the average load time for the files were 4.2±0.4 seconds and 5.4±0.2 seconds, respectively.

4.2 Module 2: Object mapping for camera localization

Object mapping is an offline phase performed by the museum staff. The purpose of this module is to generate and save a map that will be used for online localization. The map generated by ORB-SLAM is a pose graph connecting 3D map points and keyframes. Where a 3D map point consists of a 3D world position and an ORB descriptor, and a keyframe consists of an image's camera pose and ORB feature points in the image, each with its associated 3D map point. The mapping process involved capturing images of each artifact from various vantage points. As in the object recognition training phase, these vantage points were predictions of where the user will view the artifact. Thus, the map contained keyframes captured from various heights, viewing angles, and distances from the artifact, Tracking must be maintained throughout the mapping process, thus the images were processed in real-time using ORB-SLAM, as opposed to collecting images or video and post-processing the frames. The generated maps are stored onboard the mobile device upon installation of the application, thus the file size of each artifact's map was limited to a maximum of 10 megabytes. On a Google Pixel, a map with this file size had an average load time of 1.83±0.58 seconds and average processing time for the localization module of 1.85±0.69 second (Table 1).

4.3 Object Positioner Tool

The object positioner tool (Fig. 2) is used by the museum staff to calibrate the AR content's pose with respect to its corresponding map. The tool provides a menu button that allows the staff to load the artifact's AR content and map. The staff is then able to select the increment value, the type of transformation (i.e., translate, rotate, or scale), the coordinate system axis, and the direction of adjustment (i.e., positive or negative direction). The calibrated offsets for each artifact are saved. When the application is installed by the museum visitor, these offsets are stored locally on the device.

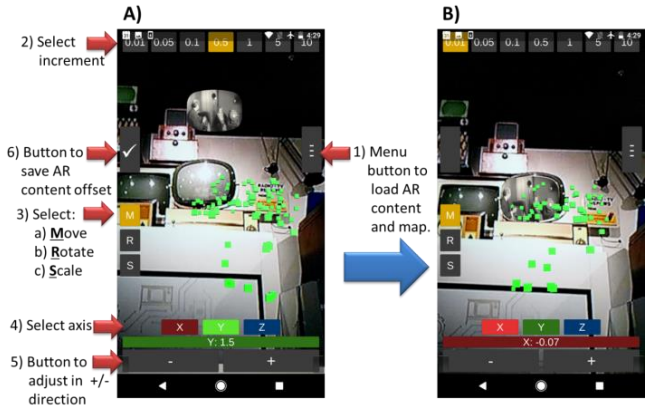

Fig. 2. A) Misalignment between the real TV screen and the AR content (TV screen). B) The Positioner Tool has been used to align the AR content with the real TV screen. The AR content's pose with respect to the map (green cubes) is stored.

4.4 Module 3: Scene recognition

Scene recognition is an online phase performed by the museum visitor. The user points the mobile device's camera at an object and DBoW2 recognizes the object, the corresponding map and AR content is then retrieved from the database, and the camera localization module is started. This module requires the vocabulary and the training database files. As previously mentioned, both files are stored locally on the mobile device when the application is installed.

4.5 Module 4: Camera localization

Camera localization is an online phase performed by the museum visitor. The user points the mobile device's camera at an object. Once Module 3 recognizes the object and retrieves the object's map from the database, this module estimates the camera pose with respect to the object (i.e., map of the object). This process allows the AR content to be displayed in the correct pose on the screen of the user's mobile device.

ORB-SLAM is used in Localization Mode, a lightweight localization algorithm used in mapped areas. In this mode the local mapping and loop closing threads are deactivated and the camera continuously localizes using the tracking thread.

5 Experiments

The AR application was built for the Android operating system. Fig. 3 shows the Cheerful Oak Stove artifact (circa 1920). When the user views the artifact using the mobile device's camera, the object recognition module automatically retrieves the corresponding AR content and map from the database. The AR content allows the user to virtually interact with the artifact by dragging logs of wood into the stove via the mobile device's touch screen. The user then places a lit match in the stove to set the wood on fire. The tracking module applies the camera localization on every frame; this allows the AR content to maintain alignment with the artifact as the user moves the camera to view the artifact from different perspectives.

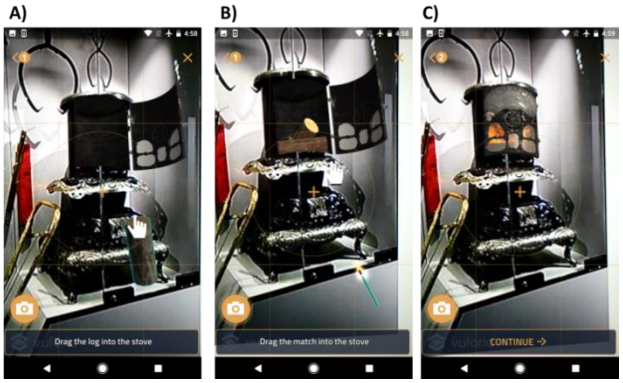

Fig. 3. Stove artifact with AR content.

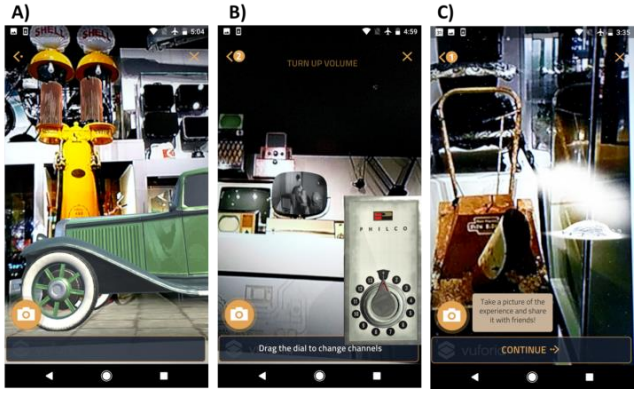

Fig. 4. A) Gas pump, B) TV, C) Snow blower.

Fig. 4 shows three other artifacts: A) A Shell twin visible gasoline pump (circa 1925), B) A Philco Predicta television (TV) (circa 1959), and C) an Outboard Marine Corporation snow blower (circa 1963). In the AR experience for the gas pump, the glass cylinders fill with fuel. A 1925 Chevrolet drives into camera's field of view. The AR experience for the television is automatically triggered by the object recognition module when the user points the mobile device's camera at the television artifact. The AR experience includes an animated television screen showing a black and white video.

There were 8 AR experiences in total. Table 1 shows results from 10 trials at each experience. Where each trial involved the user opening the application, going through the AR experience, then closing the application. The results include the average time taken for DBoW2 to recognize the object, properties of the ORB-SLAM map, the average time taken for ORB-SLAM to localize with respect to the map, and the average reprojection error upon relocalization. The reprojection error is the image distance between a projected point belonging to the 3D AR content and its corresponding 2D point in the image. 5 check points, evenly spread throughout the object, were measured to calculate each scene's reprojection error with 640x480 pixels.

5.1 Evaluation of Scene Recognition

DBoW2 performed uniformly across all experiences, with 100% precision and recall, and approximately equal recognition times. As we can see from Table 1, the average recognition time of 0.47 seconds is well within the realm of real-time performance for launch of an AR scene. We also attempted at traditional image target-based recognition with Vuforia [2], however, for almost all the artifacts, the image targets were not recognized even with slight changes in lighting or viewing angle. This demonstrates that for a practical AR application, scene recognition holds more potential for a distraction-free user experience.

5.2 Evaluation of Relocalization and Camera Motion Tracking:

Larger artifacts (e.g., the TV, the gas pump, and the Union Station lamp) required larger areas to be mapped, resulting in larger map file sizes and map load times. The experiments revealed several effects as map size increased: error in the map increased, the map points were sparser, and less ORB features were extracted from smoother images, resulting in longer localization times and larger reprojection errors (Table 1). The results also indicate that the localization time is too long for real-time AR applications. Thus, instead of invoking relocalization at every frame, it is called once to initialize the camera pose in the map's coordinate system, the device's IMU-based rotational tracker is then used to update the camera orientation. The IMU provides an orientation estimate every frame (20 Hz), however the pose drifts over time and the rotational tracker does not estimate changes in the device's position. ORB-SLAM's relocalization is called in a separate thread at 0.5 Hz, based on the results in Table 1, to provide a periodic drift correction.

Table 1. Results from 10 trials at each experience. The results include the average time taken to recognize the object, properties of the ORB-SLAM, the average time taken to localize the camera, and the average reprojection error upon relocalization.

Scene	**Recognition time (s)**	**# key frames**	**# map points**	**File size (Mb)**	**Map load time (s)**	**Localization time (s)**	**Reprojection error (pixels)**
Ray gun	0.61 ± 0.26	62	939	2.9	1.58 ± 0.52	1.83 ± 0.48	23.76 ± 13.12
Ship	0.42 ± 0.66	61	849	3.5	1.25 ± 0.48	1.55 ± 0.37	17.81 ± 10.63
Microphone	0.60 ± 0.13	64	778	2.9	1.45 ± 0.37	1.91 ± 0.81	45.88 ± 24.47
Snow blower	0.54 ± 0.30	62	982	4.7	1.82 ± 0.72	1.47 ± 0.39	36.05 ± 11.95
Stove	0.46 ± 0.25	60	884	3.2	1.28 ± 0.58	2.12 ± 0.22	30.87 ± 20.54
Gas Pump	0.35 ± 0.63	96	1095	4.1	1.91 ± 0.71	2.70 ± 0.56	52.38 ± 23.27
TV	0.23 ± 0.12	96	1338	6.5	2.54 ± 0.97	2.1 ± 0.41	69.42 ± 15.81
Lamp	0.58 ± 0.23	124	1673	8.8	2.83 ± 0.25	2.49 ± 0.31	75.23 ± 24.91
Mean	**0.47 ± 0.32**	**89.75 ± 35.16**	**1067.25 ± 299.91**	**4.56 ± 2.09**	**1.83 ± 0.58**	**1.85 ± 0.69**	**41.43 ± 16.84**

6 Conclusion

In this work, we proposed an AR framework that intelligently combines 3D scene recognition with the relocalization capabilities of SLAM to provide a smooth user experience, where AR scenes are instantly launched through scene recognition, and user's movement is tracked through SLAM-based motion tracking. We demonstrate that utilizing scene recognition results in successful AR experience design in a challenging museum environment with low light and multiple viewing angles, where traditional image target-based recognition fails. We provide detailed performance analysis of the framework with an Android application, where timings, file sizes, and error metrics for each module of the framework is presented and analyzed.
Future work involves developing a cloud processing module to offload recognition and tracking computations to a server to improve performance and provide a tool for central managing and distribution of maps for easy application administration.

References

[1] IDC, "Worldwide Semiannual Augmented and Virtual Reality Spending Guide," 2016.
[2] PCT Inc., 2019. [Online]. Available: https://developer.vuforia.com/.
[3] Google, 2019. [Online]. Available: https://developers.google.com/ar/discover/.
[4] Apple, 2019. [Online]. Available: https://developer.apple.com/arkit/.
[5] R. Mur-Artal, J. M. Montiel and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.
[6] H. Kim, S. Nagao, S. Maekawa and T. Naemura, "MRsionCase a glasses-free mixed reality showcase for surrounding multiple," ACM siggraph Asia, 2012.
[7] D. Han, X. Li and T. Zhao, "The Application of Augmented Reality Technology on Museum Exhibition—A Museum Display Project in Mawangdui Han Dynasty Tombs," International Conference on Virtual, Augmented and Mixed Reality, pp. 394-403, 2017.
[8] T. H. Tsai, C. Y. Shen, Z. S. Lin, H. R. Liu and W. K. Chiou, "Exploring Location-Based Augmented Reality Experience in Museums," Lecture Notes in Computer Science: Universal Access in Human–Computer Interaction. Designing Novel Interactions, vol. 10278, 2017.
[9] N. M. Khan *et al.*, "Towards a Shared Large-area Mixed Reality System," IEEE ICME Workshop on Mobile Multimedia Computing, pp. 1-6, 2016.
[10] B. Williams *et al.*, "A comparison of loop closing techniques in monocular SLAM," Robotics and Autonomous Systems, vol. 57, no. 12, pp. 1188-1197, 2009.
[11] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," IEEE Transactions on Robotics, vol. 28, no. 5, pp. 1188-1197, 2012.
[12] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," IEEE International Conference on Robotics and Automation, pp. 846-853, 2014.
[13] M. Shridhar *et al.*, "Monocular slam for real-time applications on mobile platforms," 2015.
[14] P. Li, T. Qin, F. Hu and S. Shen, "Monocular Visual-Inertial State Estimation for Mobile Augmented Reality," IEEE International Symposium on Mixed and Augmented Reality, 2017.
[15] A. Bonarini *et al.*, "RAWSEEDS: Robotics Advancement through Web-publishing of Sensorial and Elaborated Extensive Data Sets," Intelligent Robots and Systems Workshop on Benchmarks in Robotics Research, 2006.